RPA분류기의 성능 향상을 위한 OHC알고리즘

이 형 일†

요 약

메모리 기반 추론에서 기억공간의 효율적인 사용과 분류성능의 향상을 위하여 제안되었던 RPA(Recursive Partition Averaging)알고리즘은 대상 패턴 공간을 분할 한 후 대표 패턴을 추출하여 분류 기준 패턴으로 사용한다. 이 기법은 구성된 초월 평면상에서 단순히 대표패턴을 추출하여 분류 성능 저하의 원인이 되는 단점을 가지고 있었다. 여기에서는 기존 RPA의 단점을 보완하기 위해 FPD (Feature-based Population Densimeter)를 이용한 OHC (Optimized Hyperrectangle Carving) 알고리즘을 제안한다. 제안된 알고리즘은 RPA분할 종료 후 OHC를 이용하여 초월 평면을 최적화한 후 패턴 평균 기법을 적용하여 학습 결과를 산출한다. 제안된 알고리즘은 k-NN분류기에서 필요로 하는 메모리 공간의 40%정도를 사용하며 분류에 있어서도 RPA보다 우수한 인식 성능을 보이고 있다. 또한 저장된 패턴의 감소로 인하여, 실제 분류에 소요되는 시간비교에 있어서도 k-NN보다 월등히 우수한 성능을 보이고 있다.

OHC Algorithm for RPA Memory Based Reasoning

Hyeong-il, Lee†

ABSTRACT

RPA (Recursive Partition Averaging) method was proposed in order to improve the storage requirement and classification rate of the Memory Based Reasoning. That algorithm worked well in many areas, however, the major drawbacks of RPA are it's pattern averaging mechanism. We propose an adaptive OHC algorithm which uses the FPD(Feature-based Population Densimeter) to increase the classification rate of RPA. The proposed algorithm required only approximately 40% of memory space that is needed in k-NN classifier, and showed a superior classification performance to the RPA. Also, by reducing the number of stored patterns, it showed a excellent results in terms of classification when we compare it to the k-NN.

Key words: intelligent agent, instance-based learning, classification, machine learning

1. 서 론

최근 이용자가 취해야할 정보의 양이 방대해지고 다양해짐에 따라서 데이터 마이닝에 관한 관심이 증대되고 있다. 데이터 마이닝은 대규모의 데이터베이스로부터 유용한 정보를 추출해 내는 작업으로 일반적인 규칙을 발견하고 의사 결정 처리를 향상시키기 위해 과거의 데이터를 얼마나 효율적으로 사용하는가에 관한 문제로 볼 수 있다. 기계학습 알고리즘에서 개발되어 이러한 문제에 효율적으로 적용되고 있는 결정트리, 뉴럴 네트워크, 메모리기반 추론 알고리즘 등 다양한 알고리즘이 있다. 이러한 알고리즘 중에서 메모리 기반 추론은 주어진 문제와 가장 유사한 문제를 찾아, 미리 풀어진 문제의 해답을 적용함으로써 해답을 구하고자 하는 것이다. 여기서 메모리 기반 추론의 학습은 주어진 학습패턴 그 자체를 모두 메모리에 저장하는 것일 뿐이며, 입력패턴의 분류는 저장된 패턴들과 입력패턴사이의 거리를 이용하므로 거리기반 학습(Distance Based Learning)이라고도 한다[1,2].

접수일 : 2003년 1월 21일, 완료일 : 2003년 2월 28일
† 정회원, 김포대학 컴퓨터계열 조교수

메모리기반 학습 알고리즘에 기반을 둔 분류기로는 k-NN (k-Nearest Neighbors) 분류기를 들 수 있으며 k-NN 분류기는 메모리에 저장된 학습패턴 중 주어진 입력패턴과 가장 가까운 거리에 있는 k개의 학습패턴을 선택하여 그중 가장 많은 패턴이 소속된 클래스로 입력패턴을 분류하는 방법을 사용한다[2-4]. 이러한 k-NN 분류기는 그 성능 면에서 만족할 만한 결과를 보이고 있으며, 이미 다양한 분야에 응용되고 있다. 하지만 이 기법의 가장 큰 문제점은 학습 패턴 전체를 메모리에 저장하여야 하므로 다른 기계학습 방법에 비하여 많은 메모리 공간을 필요로 하며, 저장되는 학습 패턴이 증가할수록 분류에 필요한 시간도 많이 소요된다는 단점을 갖는다[5,6]. 따라서 메모리 기반 학습기법이 갖고 있는 문제점을 해결하기 위한 연구가 지금까지 활발히 진행되어 오고 있으며, 대표적인 연구로 IBL (Instance Based Learning)[6], NGE (Nested Generalized Exemplar)[7,8] 이론과 FPA (Fixed Partition Averaging)[10], RPA (Recursive Partition Averaging) [17]를 들 수 있다.

본 논문에서는 주어진 패턴 공간을 패턴빈도에 따라 재귀적으로 분할해 나가면서 각 분할된 초월 평면을 대표하는 패턴을 추출하여 효율적인 메모리 사용과 분류성능을 보장하는 새로운 알고리즘을 제안하고, UCI Repository의 벤치마크 데이터를 이용하여 성능을 실험적으로 검증하였다.

2. 관련 연구

2.1 k-NN 기법

k-NN 분류기는 메모리기반 학습기법을 사용한 최초의 분류기로 이 방법은 Lazy Learning Algorithm이라고도 하는데, 그 이유는 학습 시에는 단순히 학습 패턴을 메모리에 저장하며, 차후 입력패턴을 분류할 때 모든 계산이 수행되기 때문이다[11].

이러한 k-NN 분류기의 개략적인 알고리즘은 다음과 같다.

① 주어진 학습패턴을 모두 메모리에 저장한다.

② 입력패턴 $\boldsymbol{Q}$의 분류를 위하여 메모리에 저장된 모든 학습패턴과의 거리를 식 (1)을 이용하여 계산한다.

$$D_{EQ}=\sqrt{\sum_{i=1}^{n}(E_i-Q_i)^2} \quad (1)$$

이때 E는 메모리에 저장된 학습패턴을 나타내며, Q는 주어진 입력패턴이다. 또한 n은 패턴을 구성하는 특징의 개수이며, E_i, Q_i는 각각 학습패턴과 입력패턴의 i 번째 특징 값을 나타낸다.

③ 입력패턴 $\boldsymbol{Q}$와 가장 가까운 k개의 학습패턴을 선정한다.

④ 선택된 k개의 학습패턴 중 가장 많은 개수의 패턴이 소속되는 클래스로 입력패턴 $\boldsymbol{Q}$를 분류한다.

위에서 보이는 것처럼 k-NN 분류기에서의 학습은 학습패턴을 저장하는 것 이외에 아무런 조치를 취하지 않는다. 이때 k값은 분류기의 성능을 최적화하기 위하여 일반적으로 Cross Validation기법을 사용하여 결정하며, k=1인 경우를 NN 분류기라 한다[2-4]. 또한 위의 과정 중 4번째 단계에서, 입력패턴과의 거리를 이용하여 가중치를 부여하는 방법을 WeightVote k-NN 이라고 한다[3,4].

2.2 재귀 분할 평균 기법

재귀 분할 평균 (RPA: Recursive Partition Averaging) 기법은 주어진 패턴공간을 재귀적으로 분할해 나가면서 대표패턴을 추출하는 방법이다. 이 방법에서는 메모리기반 학습 기법에서 보다 효율적인 메모리 사용과 분류 성능을 보장하기 위하여 인스턴스 평균 (Instance Averaging) 법을 적용하였으며, 인스턴스 평균법은 여러 개의 학습 패턴의 특징 값을 평균하여 하나의 대표패턴으로 대치하는 방법을 말한다[14,15].

RPA는 주어진 패턴공간의 각 특징 축을 최초 2개의 영역으로 분할한다. 따라서 첫 번째 분할에서는 패턴공간이 2^n개의 공간으로 분할되며, 이때 n은 패턴을 구성하는 특징의 개수 즉, 패턴공간의 차원수가 된다. 따라서 2차원 패턴의 경우 최초 4개의 패턴공간으로 분할되며, 패턴의 분할은 현재 분할된 셀 각각에 대하여 재귀 분할 여부를 결정한다. 즉 하나의 셀에 소속되는 패턴의 클래스가 모두 같을 경우, 해당 셀의 패턴들에 대하여 패턴평균법을 적용하여 대표 패턴을 추출한다. 반대로 셀에 여러 개의 클래스에 소속되는 패턴이 혼합되어 있을 경우, 해당 셀을 다시 분할한다. 클래스가 혼합된 부분에 대해서는 점

접 세밀하게 분할해 나가게 되므로, 클래스 경계면에 위치한 셀의 경우 많은 분할이 이루어지게 된다.

그림 1에서 굵은 실선으로 표시된 부분은 실제 RPA에 의해 형성된 초월평면을 나타내는 것이며, 가는 점선으로 표시된 부분은 특징 가중치 계산을 위하여 패턴공간을 가상으로 분할한 선을 나타낸다. 이 경우 가로 특징 축은 9개, 세로 특징 축은 10개로 분할 된 것을 볼 수 있다. RPA 기법에서는 식 (3)으로 주어진 $IG(f)$ 값을 입력패턴과 메모리에 저장된 학습패턴간의 거리계산에 있어 특징의 가중치로 사용하며, 이때의 거리는 식 (4)에 의해 계산한다.

식 (2)에서 p_i는 전체 학습패턴 중 클래스 i에 소속되는 패턴의 비율, 즉 임의의 패턴이 클래스 i로 분류될 사전확률을 의미하며, C는 전체 학습패턴을 구성하는 클래스의 개수이다. 여기에서 특징 f의 가중치로 사용하는 상호정보이득 (Mutual Information Gain)은 위의 식 (3)에 계산된다[13].

RPA 기법은 특징별 패턴 분포를 고려하지 않고 단순히 클래스의 순도(purity)만을 만족시키기 위한 분할을 하고 있으며, 이것은 생성된 대표 패턴의 공간내 위치 형성에 좋지 않은 영향을 미치게 된다.

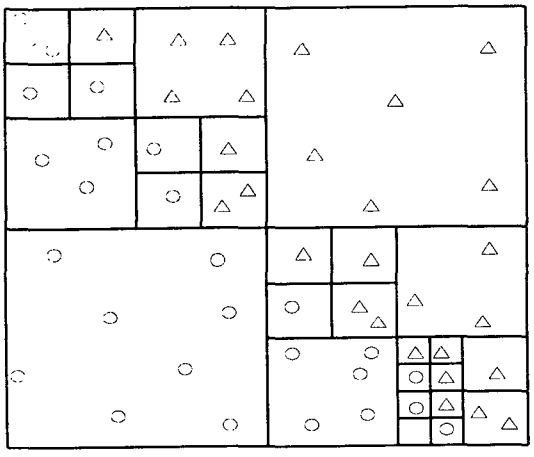

그림 1. RPA 분할

$$I = -\sum_{i=1}^{C} p_i \log_2 p_i \tag{2}$$

$$IG(f) = I - \sum_{i=1}^{N} P_i I_c \tag{3}$$

$$D_{EQ} = \sqrt{\sum_{i=0}^{n} IG(i)(E_i - Q_i)^2} \tag{4}$$

3. OHC 알고리즘

본 논문에서는 메모리기반 학습 기법에서 보다 효율적인 메모리사용과 분류 성능을 보장하기 위하여 OHC(Optimized Hyperrectangle Carving) 알고리즘을 제안하였다. OHC 알고리즘을 이용한 향상된 RPA 기법에서는 학습된 패턴을 EACH시스템에서 제안한 초월 평면 형태로 저장하는 방법을 사용한다[7]. 이 방법은 다음의 FPD를 이용해 실제 분류에 영향을 미치지 못하는 구간을 찾아 제거한 후 인스턴스 평균 (Instance Averaging) 기법을 사용하여 분류하는 방법이다.

3.1 특징의 정규화

메모리 기반 분류기에서 출력 클래스의 결정은 입력 패턴과 메모리에 저장된 학습패턴 사이의 거리를 이용하게 된다. 이 기법에서는 패턴을 구성하는 특징들이 갖는 값의 범위가 판이하게 다를 경우 문제가 발생하게 된다. 예를 들어 (0.9, 400, 0.0004), (0.8, 410, 0.02)와 같은 특징으로 구성된 패턴에서, 두 번째 특징은 다른 두 개의 특징에 비하여 상대적으로 큰 값으로 구성되어있다. 따라서 두 번째 특징이 조금만 차이가 나더라도 나머지 특징간의 차이에 관련 없이 출력 클래스가 결정된다. 이러한 문제점의 해결을 위하여 OHC 알고리즘에서는 다음의 식 5를 이용하여 특징값을 정규화한다. 이 기법은 식 5에 의하여 패턴을 구성하는 모든 특징 값을 0과 1사이의 값으로 정규화함으로써, 모든 특징의 변화가 패턴의 소속클래스 결정에 미치는 영향력을 동일하게 한다.

$$f_{i_n} = \frac{f_i - f_{i_{min}}}{f_{i_{max}} - f_{i_{min}}} \tag{5}$$

이때 f_i는 i번째 특징 값, $f_{i_{max}}, f_{i_{min}}$는 f_i가 가질 수 있는 최대, 최소 값을 나타낸다.

3.2 FPD 알고리즘

각 특징에 대한 정규화 작업이 완료되면, 공간상에 분포된 패턴들의 특징별 구간 추출 작업(이하 FPD: Feature-Based Population Densimeter)을 실행한다. 이것은 공간에 분포된 패턴의 위상(topology)를 고려하여 각 특징별 유효 구간을 형성하

는 과정이다.

FPD 알고리즘은 연속값을 가지는 특징을 고려한 연관규칙의 추출 시 사용되는 구간 분할 알고리즘과 유사하다.[18] FPD의 기본 아이디어는 연속(Continuous)된 구간을 이산(Discrete) 구간으로 분할하는 것으로 분할 구간의 크기는 각 구간에 소속된 패턴의 개수에 따라 가변 적이며, 공간상에 분포된 각 클래스 별로 별로로 이루어진다. 다시 말하면, c개의 클래스로 구성된 패턴공간의 경우 각 특징에 대하여 c번의 FPD작업이 수행된다는 것이다.

$$N_i = |\log_n(0.3 \times T_i)| \times n \tag{6}$$

$$\theta_i = AVE\left(\left|\frac{T_{i_r}}{N_i}\right|\right) \tag{7}$$

본 논문에서 제안하는 그림 2의 FPD 알고리즘은 주어진 특징에 대해 반복적으로 이진 분할해 가면서 각 구간에 식(7)과 같이 최소밀도 θ를 만족할 때까지 분할해간다. 다음으로 분할 작업이 완료되면 인접한 구간 중 최소밀도를 만족하는 구간 병합 작업을 수행한다. 이때 병합은 구간의 크기가 작은 것을 우선으로 하며, 더 이상 병합 대상이 존재하지 않을 때까지 반복된다. 여기서 클래스 i에 대한 최소밀도 θ_i는 식(7)로 계산되며, 식(6)은 특징 축의 분할 개수로 특징축을 N_i개로 분할하였을 경우, 구간에 포함되어야 하는 최소 패턴 개수이다. 또한 전체 학습패턴의 30%에 근사한 초월평면을 형성하도록 선택한 이유는, k-NN분류기에 있어 실험적으로 전체패턴의 약 30%만이 실제 분류에 사용되었다는 사실을 기준으로 한 것이다[9].

그림 3의 예제는 임계값 $\theta_i = 3$, 구간수 $N_i = 13$일때의 FPD알고리즘을 적용하는 과정이다. 이때 셀은 구간을 의미하며 셀 내부의 수치는 각 셀에 소속된 패턴의 개수를 나타낸다.

```
For all classes {
    N_i = |log_n(0.3 × T_i)| × n
    θ_i = AVE(|T_{i_r} / N_i|)
  For all features {
    FPD(N_i, θ_i)
    Record valid range R_ci
  }
}
```

그림 2. FPD 알고리즘

3.3 초월평면 최적화

RPA 분할이 완료되면, 본 논문에서는 제안된 OHC 알고리즘을 사용한다. RPA 분할 결과로 얻어진 분할된 초월 평면에서 실제 분류에 영향을 미치지 못하는 영역을 먼저 다음 그림 3과 같이 제거하고 패턴 평균법에 의한 대표패턴을 추출한다[14,15].

그림 4에서 R_{00}, R_{10}는 FPD알고리즘에 의해 생성된 유효 최빈구간을 나타낸다. 이 최빈구간은 특징의 범위 중 유효한 값을 가지는 영역을 나타내는 것으로 RPA분할에 의해 형성된 초월 평면의 최적화 작업에 사용된다. 다시 말해 형성된 초월평면 중 유효구간이 아닌 영역을 포함하는 초월 평면은 유효구간만을 포함하도록 축소하며 최종적으로 남은 구간의 패턴들을 대표패턴으로 분류 시 사용한다. 그림 4에서 회색 구간은 초월 평면 중 유효구간에 포함되지 않아 삭제 또는 축소되는 부분을 나타낸다.

4. 실험 및 분석

본 논문에서 제안한 OHC를 이용한 RPA 분류기의 성능 실험은 k-NN, RPA 기법과 비교하여 검증하였다. 실험은 기계학습의 벤치마크 자료로 사용되는 7개의 데이터를 이용하였으며, 실험 방법은 70:30 법(전체 데이터를 기준으로 70%는 학습패턴으로, 30%

Step 1	3	3	1	3	4	1	1	3	1	3	3	4	1
Step 2	6		1	7		1	1	3	3	6		4	1
Step 3	6		X	7		X	X	3	X	10			X

그림 3. FPD를 이용한 유효 최빈 구간 계산

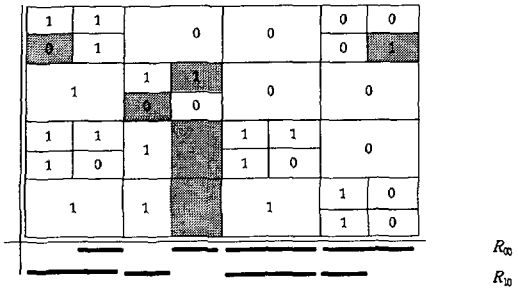

그림 4. OHC를 이용한 초월평면 최적

는 평가패턴으로 사용하는 방법)을 사용하였다[16]. 이때 70%의 학습패턴은 전체 패턴의 클래스별 분포를 고려하여 모든 클래스에서 같은 비율로 추출하였다. 실험은 Windows 2000을 적재한 PentiumIII 컴퓨터를 사용하였으며, 모든 실험결과는 25회 반복측정한 후 평균값으로 나타내었다.

4.1 실험 데이터

본 논문에서는 기계 학습의 벤치마크 자료로 사용되는 7개의 데이터를 UCI Machine Learning *Database Repository* 에서 발췌하여 사용하였으며, 이들 7개의 데이터는 Breast-Cancer Wisconsin, Glass, Ionosphere, Iris, New-Thyroid, Sonar, Wine 이며, 이들 데이터는 모든 특징이 실수 값을 갖는다. 다음의 표 1은 실험자료의 특성, 표 2는 7개의 데이터를 70:30법을 이용하여 나누었을 경우, 클래스별 학습패턴의 분포를 보여주고 있다.

표 1. 실험 데이터셋의 구성

데이터 셋	전체 패턴 개수	특징 개수	클래스 개수
Breast-Cancer Wisconsin	699	10	2
Glass	214	10	6
Ionosphere	351	34	2
Iris	150	4	3
New-Thyroid	215	5	3
Sonar	208	60	2
Wine	178	13	3

표 2. 클래스별 학습패턴 분포

데이터 셋	전체 학습패턴 개수	클래스별 학습패턴 개수					
		C1	C2	C3	C4	C5	C6
Breast-Cancer Wisconsin	488	320	168	×	×	×	×
Glass	148	53	11	0	9	6	20
Ionosphere	245	157	88	×	×	×	×
Iris	105	35	35	35	×	×	×
New-Thyroid	150	105	24	21	×	×	×
Sonar	144	67	77	×	×	×	×
Wine	123	41	49	33	×	×	×

면, RPA_{OHC} 기법의 성능이 전체 학습패턴을 고려하는 k-NN과 거의 대등한 분류성능을 보이고 있다. 또한 RPA 기법과의 비교를 보면, Ionosphere 데이터를 제외한 나머지 데이터 모두에서 RPA_{OHC}가 우수한 분류성능을 보장하고 있다.

위 실험에서 k-NN 분류기의 성능은 Leave-1-Out Cross Validation기법을 사용하여 계산한 최적의 k값을 사용한 것이며, 다음의 표 3은 각 데이터에서 사용된 k-NN 분류기의 k값을 보여주고 있다.

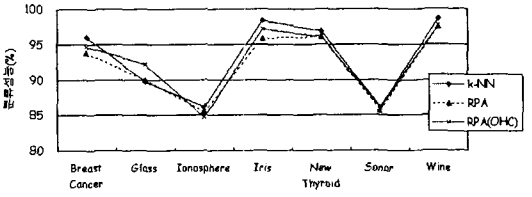

그림 5. 분류성능의 비교

표 3. k-NN 분류기의 분류성능 최적화를 위한 k값

Breast Cancer	Glass	Iono-sphere	Iris	New Thyroid	Sonar	Wine
21	1	1	51	1	1	19

4.3 메모리 사용량 비교 실험

그림 6의 실험결과에서는 k-NN, RPA, RPA_{OHC} 세 가지 방법을 이용한 분류기의, 메모리 사용량을 보여주고 있다. 결과에서 보면 k-NN의 경우 모든 학습패턴을 메모리에 저장하고 분류시 입력패턴을 모든 학습패턴과 비교한다. 하지만 RPA, RPA_{OHC}는 주어진 패턴공간을 초월평면으로 분할하여 각 초월평면을 대표하는 패턴을 저장하는 방법을 사용하며,

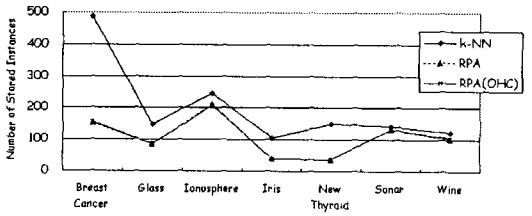

그림 6. 메모리 사용량의 비교

RPA_{OHC}는 초월 평면 형태로 저장하는 방법을 사용함으로써 우수한 메모리 사용효율을 보장하고 있다.

메모리 사용량은 전체 데이터 셋에서 RPA와 비슷한 성능을 보이고 있다. New-Thyroid 데이터의 경우 k-NN대비 약 25% 정도의 메모리만을 사용하고 있으며, Breast-Cancer 데이터의 경우는 30% 정도, Iris 데이터의 경우는 40% 정도의 학습패턴만을 메모리에 저장하는 것을 볼 수 있다. 또한 나머지 데이터에서도 약 60~80% 정도의 학습패턴만을 메모리에 저장한다. RPA와의 비교에서는 Glass, Ionosphere, New Thyroid, Sonar 4개의 실험셋에서 상대적으로 우수한 메모리 사용효율을 보장한다. 이것은 초월평면 최적화 단계에서 필요하지 않은 초월 평면을 삭제 할 수 있기 때문에 좀더 나은 메모리 사용효율을 보이게 된다.

실험 4.2와 4.3에서 보는 것처럼 본 논문에서 제안한 RPA_{OHC} 기법이 메모리 사용효율을 고려한 분류성능에 있어서 기존의 k-NN, RPA와 비교하여 우수한 성능을 보이고 있는 것을 볼 수 있다.

그림 7의 분류성능/메모리 사용량 비교에서 메모리 사용량은 k-NN분류기에서 사용하는 학습패턴의 개수를 1로 보았을 때, RPA, RPA_{OHC}의 메모리 사용량을 적용한 결과이다.

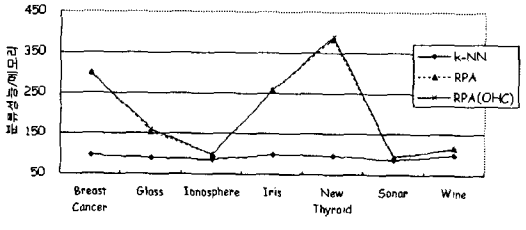

그림 7. 분류성능/메모리 사용량

4.4 분류소요시간 비교

메모리 기반 학습 기법을 이용한 분류기의 경우 메모리에 저장된 패턴의 개수와 입력패턴의 분류시간은 직접적인 관계를 가지게 되므로, 본 논문에서 제안한 최적화 초월평면을 추출하는 RPA_{OHC}기법은 실제 패턴의 분류에 소요되는 시간에 있어서도 k-NN 기법보다 월등히 빠른 분류 속도를 보장한다.

그림 8의 결과에서, RPA_{OHC}기법은 분류기 성능에 영향을 미치는 k 값을 사전에 결정하지 않고, 저장된 학습 패턴의 수를 줄임으로써 학습과 분류에 소요되는 시간이 모든 데이터 셋에 있어 k-NN기법에 비하여 월등히 적게 소요되는 것을 볼 수 있다. 결과에 표시된 값은 $\log_{10}$(소요시간)을 나타내며, 이때의 소요시간은 각 25회 반복 실험하는데 소요되는 실 소요시간(초)을 나타낸다.

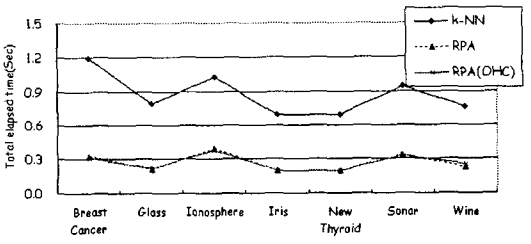

그림 8. 분류소요시간 비교

5. 결 론

본 논문에서는, 메모리 기반 추론에 있어 효율적인 메모리 사용과 분류성능을 향상시킬 수 있는 RPA_{OHC} 알고리즘을 제안하였다. 본 논문에서 제안한 RPA_{OHC} 기법은 최적 초월평면 형성을 통해 메모리에 저장되는 학습 패턴들을 초월평면으로 대체하는 방법을 채택하였으며, 실험 결과에서 볼 수 있는 것처럼 제안된 RPA_{OHC} 기법은 k-NN기법과 비교하여 모든 데이터 셋에서 적은 메모리 공간을 필요로 하며, RPA 기법과의 비교에 있어서는 실험에 사용한 벤치마크 자료 대부분에서 적은 공간을 사용하고 있다. 또한 분류 성능면에서 기존의 k-NN 기법과 거의 비슷한 성능을 보이며, RPA기법에 비해서 6개의 데이터 셋에서 우수한 성능을 보이고 있다. 마지막으로 RPA 기법에서는 분류에 영향을 미치는 패턴의 위상을 고려하지 않은 반면 RPA_{OHC}에서는 특징의 영향력과 패턴의 분포를 초월평면 형성에 반영함으로 써 좀더 우수한 분류 성능을 보장 할 수 있었다.

참 고 문 헌

[1] T. Dietterich, "A Study of Distance-Based Machine Learning Algorithms", Ph. D. Thesis, computer Science Dept., Oregon State University, 1995.

[2] D. Wettschereck and T. Dietterich, "Locally Adaptive Nearest Neighbor Algorithms", Advances in Neural Information Processing Systems 6, pp. 184-191, Morgan Kaufmann, San Mateo, CA. 1994.

[3] D. Wettschereck, "Weighted k-NN versus Majority k-NN A Recommendation". German National Research Center for Information Technology, 1995.

[4] S. Cost and S. Salzberg, "A Weighted Nearest Neighbor Algorithm for Learning with Symbolic Features, Machine Learning", Vol. 10, No. 1, pp. 57-78, 1993.

[5] D. Aha, "A Study of Instance-Based Algorithms for Supervised Learning Tasks: Mathematical, Empirical", and Psychological Evaluations, Ph. D. Thesis, Information and Computer Science Dept., University of California, Irvine, 1990.

[6] D. Aha, "Instance-Based Learning Algorithms", Machine Learning, Vol. 6, No. 1, pp. 37-66, 1991.

[7] D. Wettschereck and T. Dietterich, "An Experimental Comparison of the Nearest-Neighbor and Nearest-Hyperrectangle Algorithms", Machine Learning, Vol. 19, No. 1, pp. 1-25, 1995.

[8] S. Salzberg, "A Nearest hyperrectangle learning method", Machine Learning, no. 1, pp. 251-276, 1991.

[9] 이형일, 윤충화, "k-NN분류기의 메모리 사용과 점진적 학습에 관한 연구", 한국 정보기술전략혁신 학회, 정보과학연구 제 1권, 제1호, pp. 65-84, 1998.

[10] 정태선, 이형일, 윤충화, "고정 분할 평균기법을 사용하는 향상된 메모리 기반 추론", 명지대학교 산업기술연구소 논문지, vol. 17, 1998.

[11] D. Wettschereck, et al., "A Review and Empirical Evaluation of Feature Weighting Methods for a Class of Lazy Learning Algorithms", Artificial Intelligence Review Journal, 1996.

[12] J.R. Quinlan, "Induction of Decision Trees", Machine Learning Vol. 1, pp. 81-106, 1986.

[13] 김상귀, 이형일, 윤충화, "A study on the optimization of binary decision tree", 명지대학교 산업기술연구소 논문지, vol. 16, pp. 104-112, 1997.

[14] G. Bradshaw, "Learning about speech sounds: The NEXUS project". In Proceedings of the Fourth International Workshop on Machine Leaning, pp. 1-11, Irvine, CA: Morgan Kaufmann, 1987.

[15] T. Kohonen, "Learning vector quantization for pattern recognition (Technical Report TKK-F-A601)". Espoo, Finland: Helsinki University of Technology, Department of Technical Physics, 1986.

[16] S. Salzberg, "On Comparing Classifiers: Pitfalls to Avoid and a Recommended Approach", Data Mining and Knowledge Discovery, Vol. 1, pp. 1-11, 1997.

[17] 이형일, 정태선, 윤충화, 강경식, "재귀 분할 평균기법을 이용한 새로운 메모리 기반 추론 알고리즘", 한국정보처리학회 논문지 제6권 제7호, pp 1849-1857, 1999.

[18] 최영희, 장수민, 유재수, 오재철, "수량적 연관규칙탐사를 위한 효율적인 고빈도 항목열 생성기법", 한국정보처리학회 논문지 제6권 제10호, pp 2597-2607, 1999.

이 형 일

1985년 명지대학교 전자계산학과 졸업(학사)
1994년 명지대학교 대학원 전자계산과(석사)
2000년 명지대학교 대학원 컴퓨터공학과(박사)
1985년~1990년 (주)쌍용정보통신
1990년~1995년 (주)시에치노컨설팅
1997년~현재 김포대학 컴퓨터계열 조교수
관심분야 : 인공지능, 에이전트시스템, 정보검색
E-mail : hilee@kimpo.ac.kr

교 신 저 자

이 형 일 415-873 경기도 김포시 월곶면 포내리 산 14-1 김포대학